LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

LEVEL

0

INVENTORY

DTIC ACCESSION NUMBER

CECOM-TR-01-7

DOCUMENT IDENTIFICATION

Nov 90

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

DISTRIBUTION STATEMENT

HANDLE WITH CARE

ACCESSION FOR
NTIS GRA&I ☒
DTIC TRAC □
UNANNOUNCED □
JUSTIFICATION

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DATE ACCESSIONED

DATE RETURNED

20011101 149

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

Technical Report
HQ, U.S. Army CECOM

APPROXIMATE PROCESSING FOR REAL-TIME PROBLEM-SOLVING IN PHOENIX: A BEGINNING

Gerald M. Powell*

Center for Command, Control and Communications Systems
Communications-Electronics Command
Ft. Monmouth, NJ 07703

November 1990

CECOM-TR-01-7
Approved for public release.
Distribution is unlimited.

* The research on which the present paper is based was performed as part of the research activities of the author during a one-year Secretary of the Army Research and Study fellowhship in the Computer and Information Sciences Department at the University of Massachusetts at Amherst. The author would like to thank Paul Silvey, Dave Hart, Paul Cohen and other members of the Experimental Knowledge Systems Laboratory who provided valuable input regarding the Phoenix implementation and constructive comments regarding the ideas presented herein. Thanks are extended also to Victor Lesser who was a principal advocate of this research and to Keith Decker who provided valuable input regarding the approximate processing framework of the distributed vehicle monitoring testbed.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government endorsement or approval of commercial products or services referenced herein.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1990

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
APPROXIMATE PROCESSING FOR REAL-TIME PROBLEM-SOLVING IN PHOENIX: A BEGINNING

5. FUNDING NUMBERS

6. AUTHOR(S)
Gerald M. Powell

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Communications-Electronics Command (CECOM)
Software Engineering Center (SEC)
ATTN: AMSEL-SE-OP
Fort Monmouth, NJ 07703-5207

8. PERFORMING ORGANIZATION REPORT NUMBER
CECOM-TR-01-7

(To DTIC: October 2001)

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The research on which this paper is based was performed as part of the research activities during a one-year Secretary of the Army research and study fellowship in the Computer and Information Sciences Dept at the University of Massachusetts, Amherst.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report focuses on the problem of developing real-time problem-solving capabilities which will yield the best possible solution given the available time and computational resources. The research was part of a project called Phoenix, which aims to develop intelligent real-time planners for fighting simulated forest fires. Phoenix is designed to be a testbed for experiments in distributed control, which is one of the key characteristics of battlefield planning. The report describes research focused on: a) adapting the elements of approximate processing, which have been developed for distributed knowledge-based signal interpretation, to a planning problem having striking similarities to those observed in battlefield planning, and b) developing an approach for evaluating architectures for such real-time planning systems through the use of hypothesis testing, experiment design and statistical analysis.

14. SUBJECT TERMS Command and control; approximate processing; real time problem solving; Phoenix; modeling and simulation; planning; battlefield planning; distributed control; hypothesis testing; experiment design; architecture; statistical analysis

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

0.0 Introduction

The present paper focuses on the problem of developing intelligent real-time problem-solving capabilities which will yield the best possible solution given the available time and computational resources. The research was part of a project called Phoenix, which aims to develop intelligent real-time planners for fighting simulated forest fires.

From a high-level design view Phoenix consists of three elements: a map representation of the world which models ground-cover, elevations, natural and man-made features, and fire-state information, a discrete event simulator that coordinates the fire simulation and agent tasks, and a generalized architecture for fire-fighting agents [1].

Phoenix simulates fires in Yellowstone National Park, for which the project membership has constructed a representation from Defense Mapping Agency data. Fires spread in irregular shapes, at variable rates, determined by ground-cover, elevation, moisture content, wind velocity, and natural and man-made boundaries. Fires are fought by removing one or more things that keep them burning: fuel, heat and air. Cutting fireline, dropping water and dropping flame retardant removes fuel, heat and air respectively.

In the current Phoenix system, one fireboss directs a few bulldozers (agents), but does not control them completely. Its directions to agents specify in a coarse fashion what to do, but the agents must decide how to interpret these specifications and choose execution methods to satisfy them. Phoenix is designed to be a testbed for experiments in distributed control, which is one of the key characteristics of battlefield planning. A comparison of fire fighting and battlefield planning environments reveals many other striking similarities [2]. Because the approach employed in the present research relies heavily both on large sample sizes, in terms of problem scenarios examined, and on statistical modeling, its utility for designing intelligent agents in battlefield problem domains depends largely on the research communities' ability to develop functionally accurate simulations of those problem domains.

Phoenix agents have an architecture designed for real-time, incremental planning. This means that planners begin executing plans before planning is finished. Because the environment changes as a plan unfolds, agents must be able to monitor, anticipate plan failures, communicate with other agents, and replan. All these activities are enabled by a data structure called an envelope [1,2,4].

1.0 The Task

Inherent in fire-fighting are deadlines which fire-fighting agents must meet in attempting to minimize the loss of lives, forest and other property. Deadlines have multiple sources (externally or internally imposed on the agent who must meet the deadline) and come in different forms. For example, a fireboss could require a bulldozer (externally imposed deadline) to be at a certain location by a certain time. If a bulldozer plans its own paths, then the bulldozer may set deadlines for itself (internally imposed) in terms of how much time it is willing to spend on path planning before physically moving toward its destination.

For some problems it is impossible for an agent to find an optimal solution to a problem by a specified deadline. However, it may be possible in these situations for an agent to meet its deadline and still produce an acceptable solution by trading off some portion of the quality (e.g., accuracy) in the solution.

Implicit in the ability to make tradeoffs in the quality of a solution that will be produced in order to meet a deadline, is the availability of multiple problem-solving methods that a problem-solving agent can choose from which will result in solutions of different quality and meet different deadlines. Selecting a method, or composing a collection of methods, which will yield the most acceptable solution and meet the deadline requires knowledge of the relationship between problem characteristics (see Section 1.2) and time constraints (deadlines) and the computational costs and characteristics of solutions (accuracy of a situation assessment, total planning and acting time required, etc.) associated with different problem-solving methods.

Lesser et al. [4] have developed a framework for generating the best possible solution, with the available time and computational resources, for the task of knowledge-based signal interpretation. Lesser et al. have called their approach, which encompasses approximations in data, knowledge, and control, approximate processing.

Although problem-solving in the Phoenix domain clearly requires the interpretation of situations, it also requires plan generation, monitoring, and execution. The present research is an initial attempt to adapt the elements of approximate processing to a planning problem and to begin to expand, where it appears necessary, the approximate processing framework to address the research issues present in the Phoenix problem domain.

In evaluating the architecture of the planning component of Phoenix, with respect to the approximate processing framework developed by Lesser et al., it was determined that there were a number of differences between the Phoenix planner and its environment, on the one hand, and the distributed vehicle monitoring testbed and the approximate processing framework, on the other [5]. First, Phoenix has no notion of a goal. However, events characterized as critical in the Phoenix environment could be treated as a basis for goals. For example, if fire is located within, say, .5 miles of campers, then the goal may be to save these campers. Recognition of such a critical event could be treated as a hard deadline and the deadline could be used as a basis to decide on the priority of dealing with the particular fire. Second, since Phoenix does not do approximate processing it is necessary to develop heuristic situation-action rules. For example, if a fire is within .75 miles of campers, any approximation strategy must use only coarse methods for projecting the fire's location and for vehicle path-planning. Third, there are no approximate hypotheses in Phoenix. It will be necessary to determine what should constitute approximate hypotheses in Phoenix and develop these structures. For example, representing terrain at a less precise level would be a form of an approximate hypothesis. A variance, or uncertainty measure, could be attached to each grid in the representation and the representation could be used for projecting fire location and path-planning. Fourth, approximate knowledge sources enable processing of approximate hypotheses. We would need to determine what approximate knowledge sources would be in Phoenix. For example, there could be knowledge sources that could reason with a lower level resolution of terrain for the purposes of path-planning and fire projection. The approximate knowledge would pertain to grid properties and the measure of uncertainty for each grid. Fifth, there are no hard deadlines in Phoenix.

Presently, there are two types of deadlines in Phoenix. Figures 1 and 2 illustrate these types.

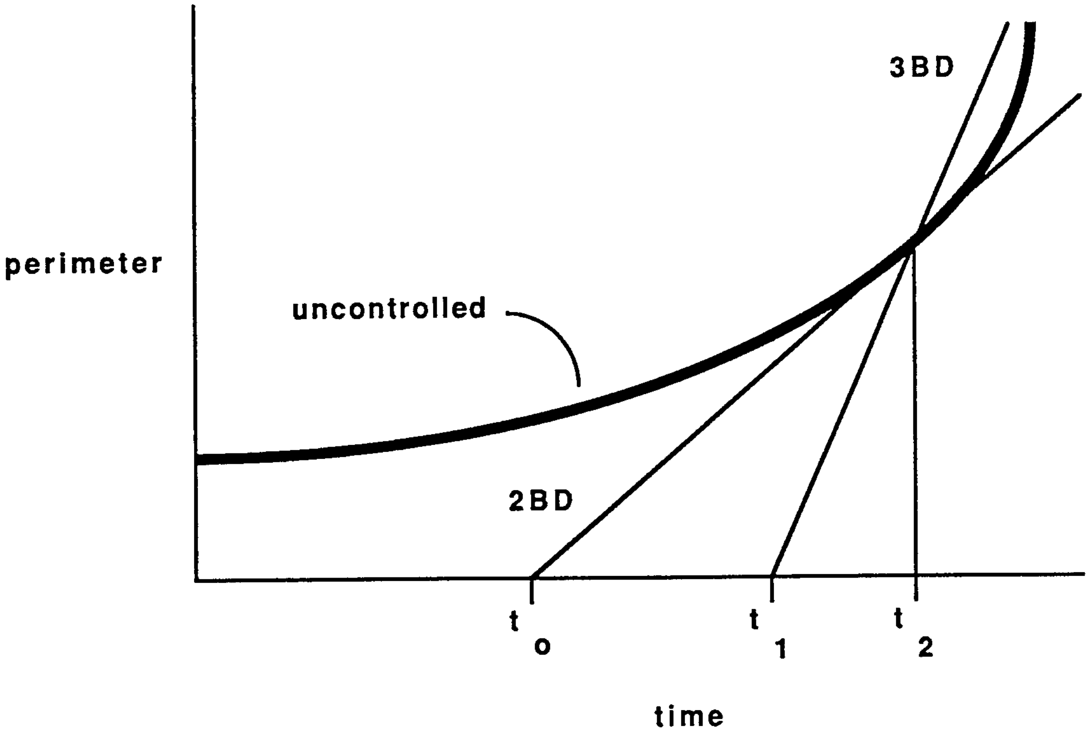

Figure 1. The relationship between the perimeter of the fire controlled vs. uncontrolled over time for a two-bulldozer (2BD) vs. a three-bulldozer (3BD) plan.

Consider the first type of deadline. An analysis of the fire provides a projection of the perimeter of it. This projection is depicted by the curve which is labelled uncontrolled in Figure 1. The dig-rate curves (BD2 and BD3) tell you when you must start digging (e.g., in three hours) in order to eventually control the perimeter of the fire. Digging must start earlier for the 2BD plan than the 3BD plan because of the difference in rates of digging. Let us suppose that at least two hours must be reserved for allowing the bulldozers to drive from their present location to the location where they should start digging fireline. Thus, there is one hour for cognitive processing time to develop a plan to tell the bulldozers which path they should take to get to the point where they should begin digging. This involves the fireboss in calculating a path; creating an envelope for the activities of the bulldozer; and telling the bulldozer to go to the point.

If there are two bulldozers involved in this plan, then paths will be needed for each. So, processing time available for each plan, assuming we must use serial processing, will be only thirty minutes each. If the estimate for computational time to do these three tasks in each plan exceeds thirty minutes, the deadline will be missed. This should invoke approximate processing in order to meet the deadline. If approximate processing fails to meet the deadline, there must be a basis for deciding what to do next. For example, if three bulldozers are available, the fireboss may consider a new plan having three bulldozers in it. If the time saved due to the higher rate of digging with a three-bulldozer plan exceeds the time lost by adding another path to plan, the three bulldozer plan will probably be preferred.

Consider the second type of deadline. The fireboss needs to calculate the path for a bulldozer's next segment. This occurs in plans called indirect attacks. In an indirect attack, the fireboss creats a polygon which provides the initial estimate for where the fireline segments should be dug. Figure 2 shows that the bulldozer is already digging line along a segment A. The fireboss does not want to let bulldozers sit idle so it should try to calculate a path before the bulldozer finishes segment A; the fireboss has an estimate of when the bulldozer should arrive at vertex AB (this relationship has an envelope on it). Waiting longer provides better situation information to inform the fireboss about where the path should be for segment B, but there is a greater chance that the bulldozer will end up idle the longer the fireboss waits to calculate the path for segment B. Calculation too early could yield a poorer result.

We need to determine the implications of using soft deadlines for approximate processing. In addition, it may be possible to have hard deadlines in Phoenix. For example, critical events could establish hard deadlines for returning a solution.

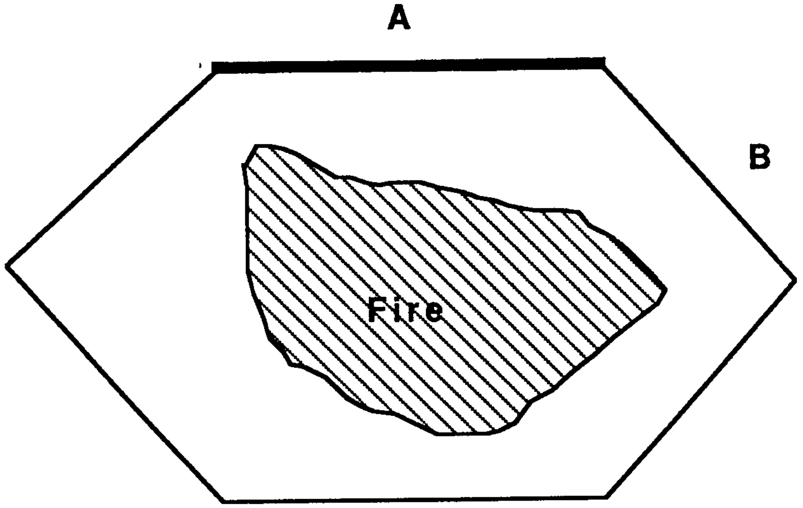

Figure 2. Fireline segments associated with an indirect attack on a fire.

The particular task in focus for this report involves getting a bulldozer from a start location to a finish location. This task can be decomposed into a path-planning subtask and a driving subtask.

Phoenix has a path-planning method that is parameterized so that it can take terrain data, as input, at five levels of resolution. The resolution is in terms of the size of the grid used to represent an area of terrain as opposed to, say, the level of abstraction of the terrain data. The polyline that the path-planning method returns varies in the density of pairs of coordinates depending on the level of resolution used; higher resolutions resulting in higher densities than lower resolutions. We refer to the combination of the path-planning method, with each of the five levels of resolution that it can take as input, as five different path-planning methods.

One of the research objectives here is to determine which characteristics of the path-planning environment (e.g., "crow-flies" distance between start and finish locations, terrain characteristics, fire-avoidance regions) differentially influence the overall performance associated with the path-planning methods.

1.1 Agent Performance

For present purposes, we define overall performance as the sum of the time required for planning and the time required for acting (driving a bulldozer in the present case; other examples could be cutting fireline or driving a fuel carrier).

$$\text{total-time} = \text{planning-time} + \text{acting-time}$$

Planning time is measured in terms of the computational time required for path planning, whereas acting time is measured in terms of the simulated time required for carrying out the action.

1.2 The Environment

For the purposes of the present study, we define environment as all of the physical and informational elements (of the world in which Phoenix fire-fighting agents sense, think, and act) that are external to the agent and that have the potential to influence the problem-solving performance of Phoenix agents. Our categorization of these elements includes terrain characteristics (e.g., wind velocity, precipitation), fire characteristics (e.g., estimated size of perimeter initially observed, location of fire with respect to terrain type), task specifications (e.g., cut fireline from location A to location B; the distance between A and B is expected to impact the time for both planning a fireline segment and cutting line along that segment), operational constraints (e.g., the availability of physical resources of different types, in different quantities, and at different times; or an externally imposed order in which tasks must be accomplished), spatial constraints (e.g., restricted geographic areas such as lakes and fire avoidance regions), temporal constraints (deadlines), and information characteristics (e.g., completeness, accuracy, and timeliness).

Knowledge of the relationship between these characteristics and the performance profiles associated with the path-planning methods can be used as a basis for selecting a method for path planning that is likely to achieve the specified objective (driving a bulldozer from location A to location B) by the specified deadline. Knowledge of this relationship can also be used in composing approximate processing strategies in which path planning, and the physical activities associated with it (e.g., driving), is only one of the numerous requirements one may need to satisfy in an overall fire-fighting problem.

1.3 Purpose of the Study

This initial study has two major purposes. One purpose is to understand the relationship between features of the environment and the performance measures. The relationships will be expressed in the form of statistical models. A second purpose is to use the knowledge we acquire (reflected in the models), from examining the relationship between environmental features and performance, to guide the design of a situation assessment element and to evaluate the utility of that element in the bulldozer transit task.

With respect to the first purpose of this study, there are certain features of the path-planning problem (environment) that influence which partial paths get expanded during path planning. So, in addition to influencing the location and length of the path that gets generated, they influence path-planning time. These features are: start-location, finish-location, non-crossable rivers, ground cover (hardwood and softwood), FARs, lakes and roads. In certain scenarios the influence of these features seems obvious. For example, if the grids between the start and finish locations are either all softwood or all hardwood, we find path-planning time for these two cases to be equal but driving time should be longer for the hardwood case than the softwood; these findings should be observed independently of the path-planning method used. In the case where the type of ground cover between the start and finish locations is heterogeneous, and the path-planning method uses a grid size that is the same as the level used for the Phoenix terrain reprsentation (256 meters/side), we should find that the bulldozer drives exactly the path given to it and, therefore, requires the same driving time as that calculated by the planner. We should not expect this result when the path-planning method uses a grid size larger than that used for the terrain represenation. For example, there may be grids with hardwood, in sections of the path the bulldozer is given, which were missed during path-planning due to the grid size used. Consequently, in this case, the bulldozer should take longer, than the driving time calculated by the path-planner, to reach its finish location. Whether it actually will take longer to reach its destination, and by how much, needs to be determined empirically.

With regard to the second major purpose of the present study, the initial situation assessment element will generate a characterization, of each instance of the bulldozer transit task, in terms of the relevant features of the environment (see Section 2.1.3 for the initial set of predictor variables). This characterization will be used to select the most appropriate path-planning method for accomplishing the bulldozer transit task. The situation assessment element will be empirically examined initially to measure its utility in increasing overall performance in this task. It is anticipated that this element will have utility at least for other ground-based transit or digging tasks.

2.0 Approach

2.1 Assessment of the Environment and Performance

The approach we will take to try to understand the relationship between environmental features and the performance measures (all variables are defined in Appendix A) is the following:

2.1.1 The five path-planning methods (resolution in meters/side: 256, 512, 1024, 2048, 4096) will be run, for 100 fires each, in the Phoenix demonstration scenario. Each of the five methods will plan against each different fire until ten paths have been collected or the termination condition (a time limit) has been reached. The combination of methods/fire, fires, and paths/method will yield a total of approximately 5,000 paths. During each fire, data will be collected on the following variables:

* start-location
* finish-location
* path-planning-method
* path-planning-time
* driving-time
* total-time
* path-generated
* fire-size-est

Each time a different fire is set, a value will be randomly chosen (and recorded), at the beginning of the trial, for each of the following variables:

* fire-size-actual
* fire-location
* windspeed (held constant within a trial)
* wind direction (held constant within a trial)

2.1.2 After all of the above data are collected, values will be determined for the following variables for each trial:

* pos-y-dev
* neg-y-dev
* overall-y-dev
* mean-y-dev
* sd-y
* ucl-y
* lcl-y

* x-dev-start
* mean-x-dev-start
* sd-x-start
* lcl-x-start

* x-dev-finish
* mean-x-dev-finish
* sd-x-finish
* ucl-x-finish

* crow-flies-distance
* planned-path-length
* %polyline-road

* non-crossable-river-in-PER
* %hardwood-PER
* %softwood-PER
* %FAR-PER
* %lake-PER
* %road-PER

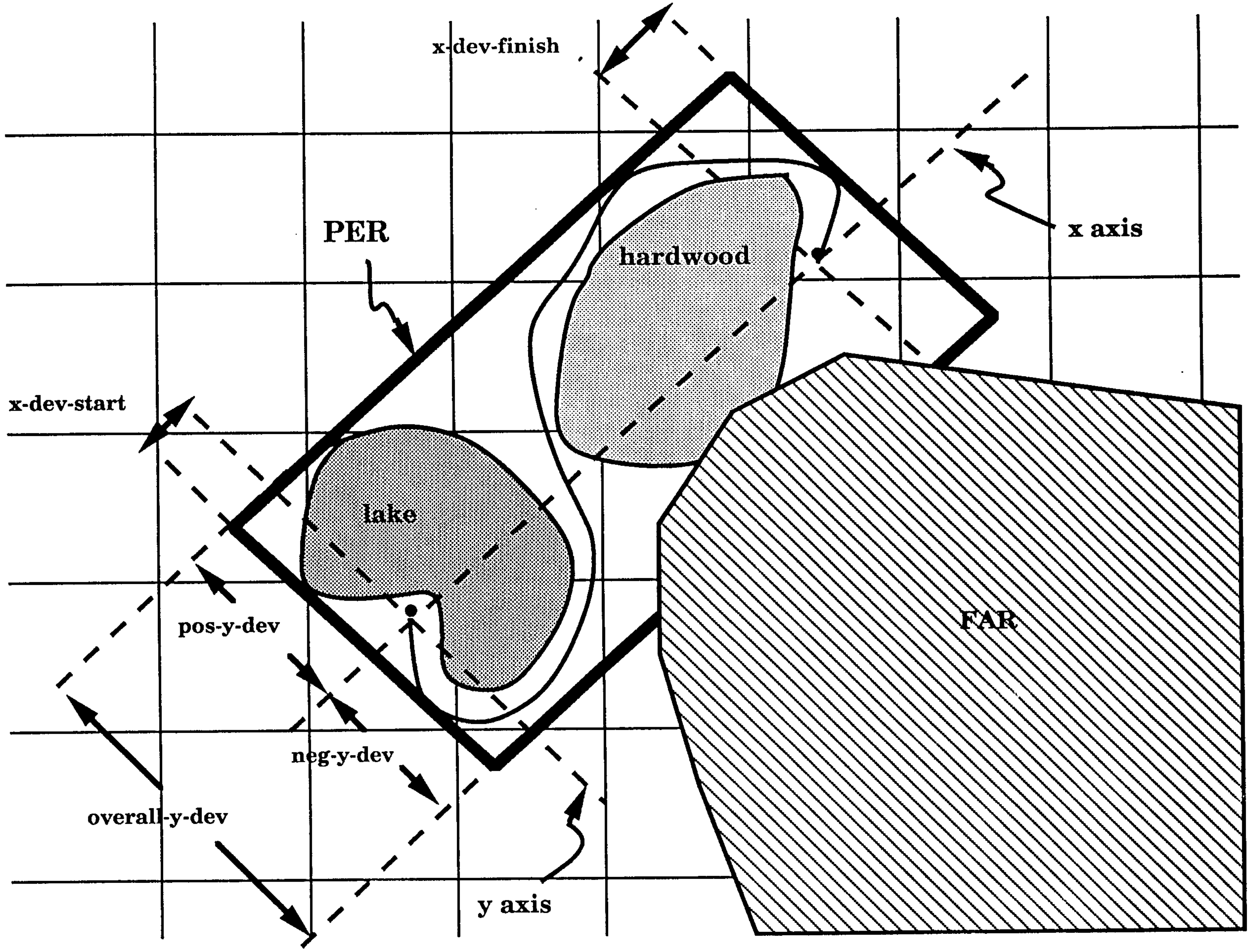

Figure 3. An example of a path-extent-region (PER) depicted for a planned path.

For the last group of variables (beginning with non-crossable-river-in-PER), their values will be obtained by iterating over every grid in the path-extent-region (PER) for every trial. The iteration will involve the Phoenix terrain model (256 meters/side), and the representations for FARs, lakes, non-crossable rivers, and roads.

Figure 3 illustrates the elements involved in defining the PER. For each trial a path is calculated from start-location to finish-location. Crow-flies distance is based on a line drawn from the start-location (the point to the left of the lake) to the finish-location (the point to the right of the hardwood area). This line will represent the x axis as shown in Figure 3.

The coordinates that represent the maximum deviation of the path along the y axis, in the positive direction (pos-y-dev) and, in the negative direction (neg-y-dev), provide a basis for calculating each path's overall deviation with respect to the x axis (overall-y-dev). Calculations of overall-y-dev will be grouped by path-planning-method, i.e., there will be five groups of data for overall-y-dev.

Based on the five groupings of data obtained for overall-y-dev and the number of trials run for each group, the mean (mean-y-dev), standard deviation (sd-y), upper confidence limit of y (ucl-y), and lower confidence limit of y (lcl-y) will be calculated.

Calculations of the aspect of the PER corresponding to deviations along the x axis will be done as follows. For each trial, the maximum negative deviation along the x axis, from the starting location (x-dev-start) and the maximum positive deviation along the x axis from the finish location (x-dev-finish) will be measured. Means and standard deviations based on these two measures, mean-x-dev-start, mean-x-dev-finish, and sd-x-start, sd-x-finish, respectively, will be calculated. Then the lower confidence limit of the x coordinate at start-location and the upper confidence limit of the x coorrdinate at finish-location, lcl-x-start and ucl-x-finish, respectively, will be calculated.

If the distributions of overall-y-dev, x-dev-start, and x-dev-finish are normal, we can be approximately 68%, 86%, 95%, and 99% confident that a path will be planned in the region defined by the values of ucl-y, lcl-y, lcl-x-start, and ucl-x-finish associated with each path-planning method. We call this region the probable search region (PSR). The four different levels of confidence are associated with the use of a standard deviation value of 1.0, 1.5, 2.0, or 2.5.

2.1.3 Three multiple linear regressions (MLRs) will be applied to these data. The predictor variables will be:

* crow-flies-distance
* non-crossable-river-in-PER
* %HW-PER

* %SW-PER
* %FAR-PER
* %lake-PER
* %road-PER
* path-planning-method

The criterion variables will be:

* path-planning-time
* driving-time
* total-time

The results from the MLRs will provide a basis for decisions about further analyses on the data collected. For example, if path-planning-method and %HW-PER account for a significant proportion of the variance in performance, it is likely an analysis of variance (ANOVA) will be performed on these variables (%HW-PER would need to be decomposed into intervals of values).

2.1.4 The use of confidence intervals of different sizes in determining the PSR (Section 2.1.2) provides another mechanism for attempting to meet deadlines.

After the four confidence limits (ucl-y, lcl-y, lcl-x-start, ucl-x-finish) are calculated for each of the four confidence intervals (standard deviations = 1.0, 1.5, 2.0, or 2.5), means and standard deviations will be calculated for total-time associated with each of these four intervals. These statistics will be used to provide estimates of this performance measure. These time estimates provide another variable to be considered in attempting to meet deadlines. Of course, associated with the choice of a particular confidence interval is a probability of failure. For example, only the 68% and 86% confidence intervals may have performance times associated with them that meet the deadline; let's assume the performance time is shorter for the former than the latter. However, in the case of the former, there is a 32% chance that no path will be found that meets the deadline whereas there is only a 14% chance of this occurring with the latter. A one-way ANOVA will be performed on total-time to determine if there is a significant difference in this measure due to confidence interval size.

By explicitly representing a level of confidence associated with the occurrence of an outcome (e.g., total-time) within a particular interval of values, confidence intervals provide a basis for a decision-maker to trade off different levels of risk with different levels of expected performance.

3.0 Model

3.1 Environmental Influences on Path-planning and Driving

As mentioned earlier, one major objective of this study is to collect data, during fire-fighting scenarios, on various environmental features and bulldozer performance in terms of path-planning and driving times. In this manner we hope to discover the features of the environment that significantly influence performance in the bulldozer transit task.

Multiple linear regressions will be performed on these data in an attempt to determine the variance in performance accounted for by the environmental features examined. The findings from the MLRs will be used as a basis for statistically evaluating the differences in performance associated with the different levels of predictor variables. The end product of these analyses will be a statistical model reflecting relationships between environmental features and performance in the bulldozer transit task.

3.2 Situation Assessment for Path-planning and Driving

The findings from the MLRs, which provide a statistical model of environmental influences on performance, and our knowledge of PSRs (expressed in the form of statistical models, Section 2.1.2), will be used to guide the design of a situation assessment element to be used, by the Phoenix system, in conjunction with the path-planning methods.

In addition to designing a situation assessment element, the second major objective of the present study is to evaluate the utility of this element in terms of its impact on overall performance in the bulldozer transit task.

4.0 Prediction

Using our understanding of the path-planning method (i.e., we can clearly identify certain environmental features which are considered by the method during path-planning), we expect at least the following hypotheses to be worth testing with respect to the first major objective of the present study.

4.1 Hypotheses:

H1: When grid size (used by the path-planning method) increases, path-planning time should decrease

H2: When terrain between start and finish locations is homogeneous, driving time associated with each of the five methods should be equal

H3: When terrain between start and finish locations is heterogeneous, driving time associated with each of the five methods should not be equal

With respect to the second major objective of the present study we expect at least the following hypotheses to be worth testing.

4.2 Hypotheses:

H1: Use of a situation assessment element in the present task will result in less computation time overall (i.e., situation-assessment-time, planning-time, and driving-time) than performing the task without this type of element.

5.0 Design and Implementation

At the present time the studies required for collecting data on the relationship between environmental factors and performance in the bulldozer transit task have not been completed. Once collected, MLRs will be performed on these data in order to develop a statistical model that reflects the relationship between the predictor and criterion variables. Further statistical analyses (e.g., ANOVA) will be used to develop a more refined statistical model.

Having completed the studies mentioned above, we will have the information necessary to construct an initial design specification, for the situation assessment element, based on the various statistical models, and desired performance requirements.

It is assumed that different levels of abstraction of terrain data will exist. These abstractions could incorporate geographical boundary points (e.g., north, south, east, west) to represent the spatial extent of terrain features such as lakes and non-crossable rivers. Fire avoidance regions could also include such boundary points in their representation. Different levels of approximate data/hypotheses should be made available for attempting to meet different deadlines.

Probable search regions will allow the use of confidence intervals of different sizes which will provide a basis for attempting to meet different deadlines. It is envisioned that the situation assessment element might be designed as shown in Figures 4 and 5.

Figure 4 is a flowchart associated with a preliminary design for a situation assessment element for path-planning in Phoenix. Situation assessment is initiated with the following input provided to Identify Viable PPMs (path-planning methods): start location and finish location for driving from start to finish; a deadline for classifying the situation and returning the most appropriate PPM; a deadline for driving from the starting location to the finishing location. It should be noted that the process reflected in Figure 4 should include an element that monitors the time remaining for situation assessment activities so as to avoid missing the deadline associated with situation assessment.

In Figure 4 the element Identify Viable PPMs is intended to perform the following functions. Calculate the distance between the starting and

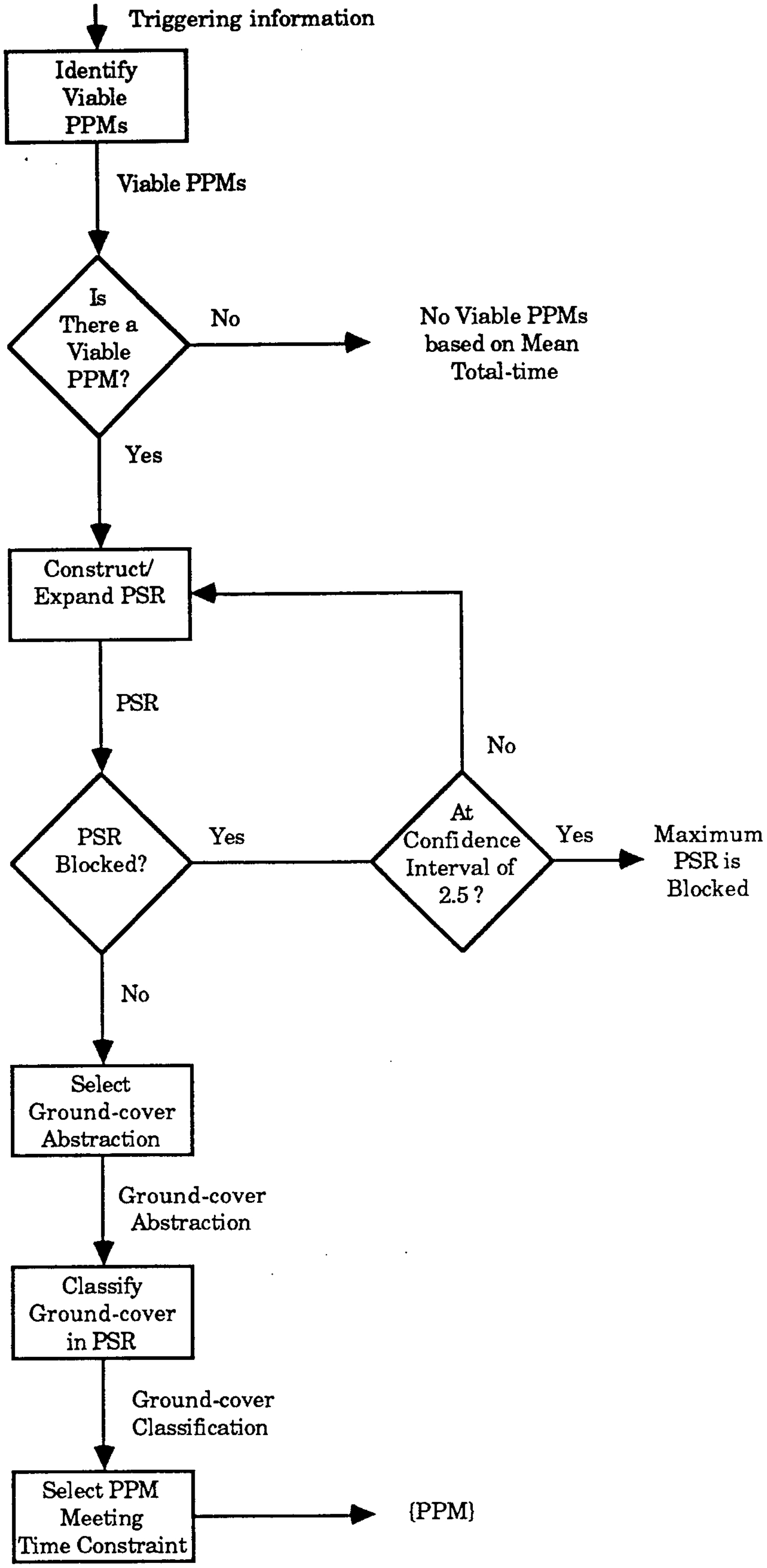
Triggering information
Identify Viable PPMs
Viable PPMs
Is There a Viable PPM?
No
No Viable PPMs based on Mean Total-time
Yes
Construct/ Expand PSR
PSR
No
PSR Blocked?
Yes
At Confidence Interval of 2.5 ?
Yes
Maximum PSR is Blocked
No
Select Ground-cover Abstraction
Ground-cover Abstraction
Classify Ground-cover in PSR
Ground-cover Classification
Select PPM Meeting Time Constraint
{PPM}

Figure 4. A depiction of control and data flow for the situation assessment element. PPM=path planning method; PSR=probable search region

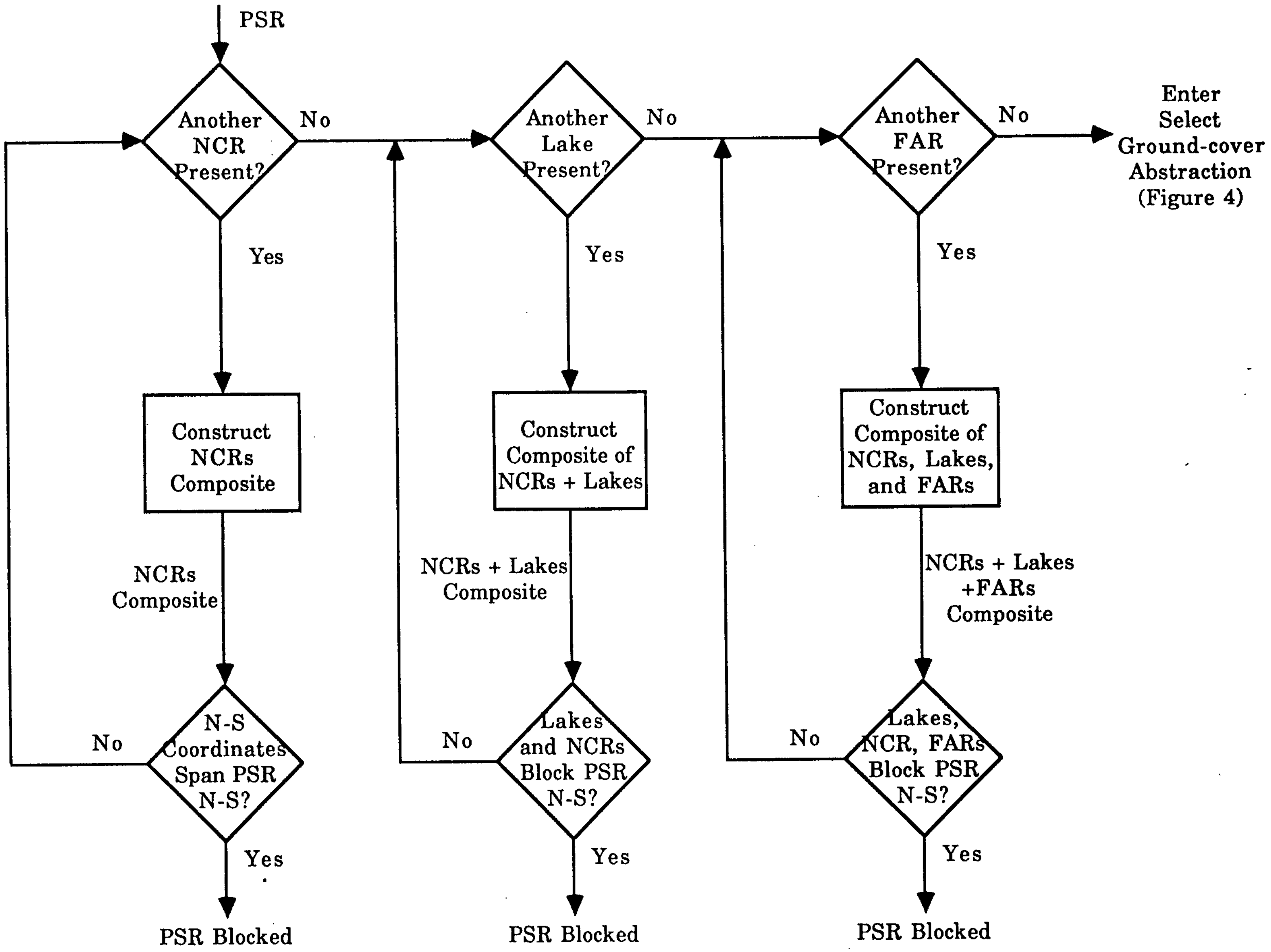
PSR
Another NCR Present?
No
Yes
Construct NCRs Composite
NCRs Composite
N-S Coordinates Span PSR N-S?
No
Yes
PSR Blocked
Another Lake Present?
No
Yes
Construct Composite of NCRs + Lakes
NCRs + Lakes Composite
Lakes and NCRs Block PSR N-S?
No
Yes
PSR Blocked
Another FAR Present?
No
Enter Select Ground-cover Abstraction (Figure 4)
Yes
Construct Composite of NCRs, Lakes, and FARs
NCRs + Lakes +FARs Composite
Lakes, NCR, FARs Block PSR N-S?
No
Yes
PSR Blocked

Figure 5. This figure expands the PSR Blocked? element (Fig. 4) to depict the details of the process. NCR=non-crossable-river; N-S=north-to-south; FAR=fire avoidance region; PSR=probable search region

finishing locations. Substract specified situation assessment time from total-time (planning- and driving-time). Enter the data structure (see Table 1) at the appropriate distance interval and prune all methods not having a mean total-time less than the specified required total time. If at least one PPM remains, use 1.0 as the confidence interval size.

To be viable, a PPM must have a mean total-time which is less than the required total-time specified. If no viable PPMs are identified, the Phoenix element invoking situation assessment should be notified. Alternatively, the criterion for meeting the total-time requirement could be relaxed by some quantity, say, one standard deviation and the analysis could be repeated on that basis. If at least one viable PPM is eventually identified, then construct a PSR using the associated confidence interval size (see Construct/Expand PSR in Figure 4). Next, analyze the PSR for the existence and location of non-crossable rivers, lakes, and FARs in that order as shown in Figure 4 (PSR Blocked?) and in Figure 5. If these three features, in any combination, prevent traversal of the PSR, the next largest confidence interval is used and the analysis of the PSR for these features is repeated. We would like to record results from analyses of the PSR and use them in each successive analysis to determine whether the enlarged PSR contains a path which allows traversability from the start to finish locations.

If the PSR is found to be not traversable after all available confidence intervals are considered, the Phoenix element that invoked situation assessment is notified. Otherwise, the first PSR which contains a traversible region will be used for further analysis. A choice is made for a particular abstraction level for the ground-cover representation (Select Ground-cover Abstraction - Figure 4). This choice could be guided by time remaining for situation assessment. The ground-cover in the PSR would then be classified according to a classification scheme determined a priori. Using the knowledge of the relationship between the PPMs and ground-cover classes, and the data on total-time (planning and driving times) associated with the various pairwise combinations of PPM and ground-cover class, the PPM providing the shortest total-time is selected and this information is provided to the Phoenix element that invoked situation assessment.

It should be noted that in this description of a design of a situation assessment element the presence of roads in the PSR is not considered. Clearly, we would want a situation assessment element to evaluate the PSR for the presence of roads and to use that information as part of the situation assessment.

6.0 Experiments

6.1 Method

Distance (miles)		Confidence Interval (+ and -)	Total-time (Mean + SD)
PPM-i	< .5	1.0	
	< .5	1.5	
	< .5	2.0	
	< .5	2.5	
	.5 - 2.0	1.0	
	.5 - 2.0	1.5	
	.5 - 2.0	2.0	
	.5 - 2.0	2.5	
	2.1 - 4.0	1.0	
	2.1 - 4.0	1.5	
	2.1 - 4.0	2.0	
	2.1 - 4.0	2.5	
		⋮	
PPM-i+n	< .5	1.0	
	< .5	1.5	
	< .5	2.0	
	< .5	2.5	
	.5 - 2.0	1.0	
	.5 - 2.0	1.5	
	.5 - 2.0	2.0	
	.5 - 2.0	2.5	
	2.1 - 4.0	1.0	
	2.1 - 4.0	1.5	
	2.1 - 4.0	2.0	
	2.1 - 4.0	2.5	

Table 1. The total-time values (means + standard deviations) associated with the various combinations of path-planning method (PPM), distances from start to finish locations, and confidence interval size.

We will evaluate the utility of the situation assessment element in terms of overall-time which is obtained by summing the following measures:

- * situation-assessment-time
- * path-planning-time
- * driving-time

The experimental design here involves comparing overall-time, on a large number of trials for different scenarios, when the situation assessment element is used versus when it is not.

6.2 Results

Statistical analysis for this experiment will provide a basis for deciding whether the initial situation assessment element design improves performance in the task or not. Whether the initial design results in improved performance or not, other designs for a situation assessment element can be specified and evaluated.

7.0 Discussion

7.1 Redesign

Decisions about subsequent designs for the situation assessment element should be guided by the performance data collected to evaluate earlier designs. This suggests that timing data should be collected on the different processing steps of situation assessment. The data may indicate, for example, that most of the computation time is being spent on a particular aspect of the situation assessment and that modifications in terms of, say, new approximations may reduce computation time.

8.0 References

[1] Cohen, P.R., Greenberg, M.L., Hart, D.M. and Howe, A.E. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, Fall 1989.

[2] Powell, G.M. and Cohen, P.R. Operational planning and monitoring with envelopes. *Proceedings of the IEEE Fifth Annual AI Systems in Government Conference*, May 1990.

[3] Hart, D.M., Anderson, S. and Cohen, P.R. Envelopes as a vehicle for improving the efficiency of plan execution. *University of Massachusetts, Department of Computer and Informations Sciences,Technical Report 90-21*, 1990.

[4] Lesser, V.R., Durfee, E.H. and Pavlin, J. Approximate processing in real-time problem solving. *AI Magazine,* Spring 1988.

[5] Powell, G.M. Presentation, Experimental Knowledge Systems Laboratory, Computer and Information Sciences Department, University of Massachusetts, March 1990.

9.0 Appendix A

start-location-x: the starting x-coordinate the path-planning method will use in planning a path

start-location-y: the starting y-coordinate the path-planning method will use in planning a path

finish-location-x: the finishing x-coordinate the path-planning method will use in planning a path

finish-location-y: the finishing y-coordinate the path-planning method will use in planning a path

path-planning method: the path-planning method used (one of five) in each trial

path-planning time: the computational time required to plan a path

driving-time: the simulated time taken to drive from the start-location to finish-location

situation-assessment-time: the computational time required to perform a situation assessment

total-time: path-planning-time + driving-time

overall-time: situation-assessment-time + total-time

path-generated: a record of the path (polyline) planned

fire-size-est: this is the initial estimate of the perimeter of the fire

fire-size-actual: this is the size of the fire set at the beginning of a trial

fire-location: a record of the location of the fire (denoted by several grid coordinates at the perimeter of the fire ??)

wind-speed: the speed of the wind; randomly selected at the beginning of each different fire (held constant within a trial)

wind-direction: the direction of the wind (denoted by bearing in terms of degrees, or what ??)

non-crossable-river-in-PER: a predicate that evaluates to true or nil; it evaluates to true if, upon iterating over the terrain in the PER, a non-crossable river is detected in any grid in the PER [note: iteration over PER will be done at grid size = 256 meters/side]

crow-flies-distance: the straight line (aerial) distance between start-location and finish-location

planned-path-length: the length of the path (polyline) generated during each run of a path-planning method

%polyline-road: the percentage of the path (polyline) generated that is occupied by road

%hardwood-PER: the percentage of hardwood in the path-extent region

%softwood-PER: the percentage of softwood in the path-extent-region

%FAR-PER: the percentage of fire avoidance region (FAR) in the path-extent-region

%lake-PER: the percentage of lake in the path-extent-region

%road-PER: the percentage of road in the path-extent-region

pos-y-dev: the positive deviation along the y-axis perpendicular to the line between the y coordinate at the start location and the y coordinate at the finish location

overall-y-dev: [y - (-y)], i.e., [pos-y-dev - (neg-y-dev)]

mean-y-dev: mean of the values obtained for overall-y-dev

sd-y: standard deviation of values obtained for overall-y-dev

ucl-y: upper confidence limit of y; mean-y-dev + s * (sd-y), where s = 1.0, 1.5, 2.0, or 2.5

lcl-y: lower confidence limit of y; mean-y-dev - s * (sd-y), where s = 1.0, 1.5, 2.0, or 2.5

x-dev-start: the negative deviation along the x-axis from the start location

mean-x-dev-start: mean of the values obtained for x-dev-start

sd-x-start: standard deviation of values obtained for x-dev-start

lcl-x-start: lower confidence limit of x coordinate at the start
location; mean-x-dev-start - s * (sd-x-start), where
s = 1.0, 1.5, 2.0, or 2.5

x-dev-finish: positive deviation along x-axis from the finish location

mean-x-dev-finish: mean of the values obtained for x-dev-finish

sd-x-finish: standard deviation of values obtained for x-dev-finish

ucl-x-finish: upper confidence limit of x coordinate at the finish
location; mean-x-dev-finish + s * (sd-x-finish), where
s = 1.0, 1.5, 2.0, or 2.5